Scalable Neural Decoders for Practical Fault-Tolerant Quantum Computation

Andi Gu [1] **J. Pablo Bonilla Ataides** [1] **Mikhail D. Lukin** [1] **Susanne F. Yelin** [1]

Abstract

Fault-tolerant quantum computing, the prerequisite for every scalable quantum algorithm, hinges on a classical decoder that is simultaneously fast and accurate. Existing decoders face a sharp accuracy–latency trade-off: iterative decoders are fast but too inaccurate on high-rate codes, while near-optimal combinatorial-search decoders are far too slow for real-time use. We treat decoding as a structured inference problem on a spatiotemporal graph with known symmetry, and introduce Cascade: a translation-equivariant convolutional decoder that applies the same architectural template to both surface codes and high-rate quantum LDPC codes, adapting only the convolution to the code's lattice symmetry. A single model trained at one high physical error rate generalizes across seven orders of magnitude in logical error rate with well-calibrated uncertainty, and error suppression improves continuously with model capacity. On the $[\![144, 12, 12]\!]$ Gross code at $p = 0.1\%$, Cascade reaches a logical error rate of $\sim 10^{-10}$, up to $\sim 4000\times$ lower than widely-used iterative decoders (Roffe et al., 2020) and $\sim 17\times$ lower than the strongest iterative variant (Müller et al., 2025), while matching the accuracy of a near-optimal combinatorial decoder (Beni et al., 2025) at 3–5 orders of magnitude lower latency. This accuracy reveals a previously inaccessible "waterfall" regime of error suppression, substantially steeper than current scaling assumptions (Gidney & Ekerå, 2021; Beverland et al., 2022) and implying lower space-time overhead for fault-tolerant quantum computation than previously anticipated.

[1]Department of Physics and Harvard Quantum Initiative, Harvard University, Cambridge, MA, USA. Correspondence to: Andi Gu <andigu@g.harvard.edu>.

Accepted at ICML 2026 AI for Science Workshop.

1. Introduction

Decoding quantum error-correcting codes is a large-scale structured prediction problem: from binary detection events produced by repeated stabilizer measurements on a spatiotemporal lattice, infer the equivalence class of the underlying physical error. The problem has rich geometric structure (detection events live on fixed sites of a regular lattice, and the parity-check structure is invariant under the lattice's symmetry group), but existing solvers fail to exploit this fully. Iterative belief propagation (BP) is fast but suffers from well-studied convergence failures that fundamentally limit its accuracy on quantum codes (Poulin & Chung, 2008; Raveendran & Vasić, 2021); combinatorial search methods achieve near-optimal accuracy but require up to a second per shot (Beni et al., 2025), far outside the real-time decoding budgets of quantum hardware (Bluvstein et al., 2024). The gap is particularly acute for the new generation of high-rate quantum low-density parity-check (LDPC) codes (Leverrier & Zemor, 2022; Panteleev & Kalachev, 2022; Bravyi et al., 2024), whose complex stabilizer structure is essential for efficient fault tolerance but does not admit the graph-matching reduction that makes surface codes tractable (Fowler et al., 2012; Higgott, 2022). Decoder accuracy is not a secondary concern: fault-tolerant algorithms (Gidney & Ekerå, 2021; Campbell et al., 2017; Beverland et al., 2022) demand logical error rates of 10^{-10}–10^{-12}, and current hardware platforms (Ballance et al., 2016; Li et al., 2024; Evered et al., 2025) have reached physical error rates where the decoder's accuracy is the dominant factor in whether those targets are within reach.

Combining the right physical inductive biases (locality, lattice symmetry, anisotropy) with sufficient model capacity can expose features of the decoding problem that hand-designed algorithms are too inaccurate to see. Here, that feature is a waterfall regime of error suppression standard decoders miss (Section 4). We introduce Cascade, a neural decoder whose architecture is built around the geometric regularity of quantum codes. For any code whose stabilizers share a common local structure up to lattice translations (a class that includes surface codes, bivariate bicycle (BB) codes (Bravyi et al., 2024), color codes (Bombin & Martin-Delgado, 2006), and other quantum LDPC codes

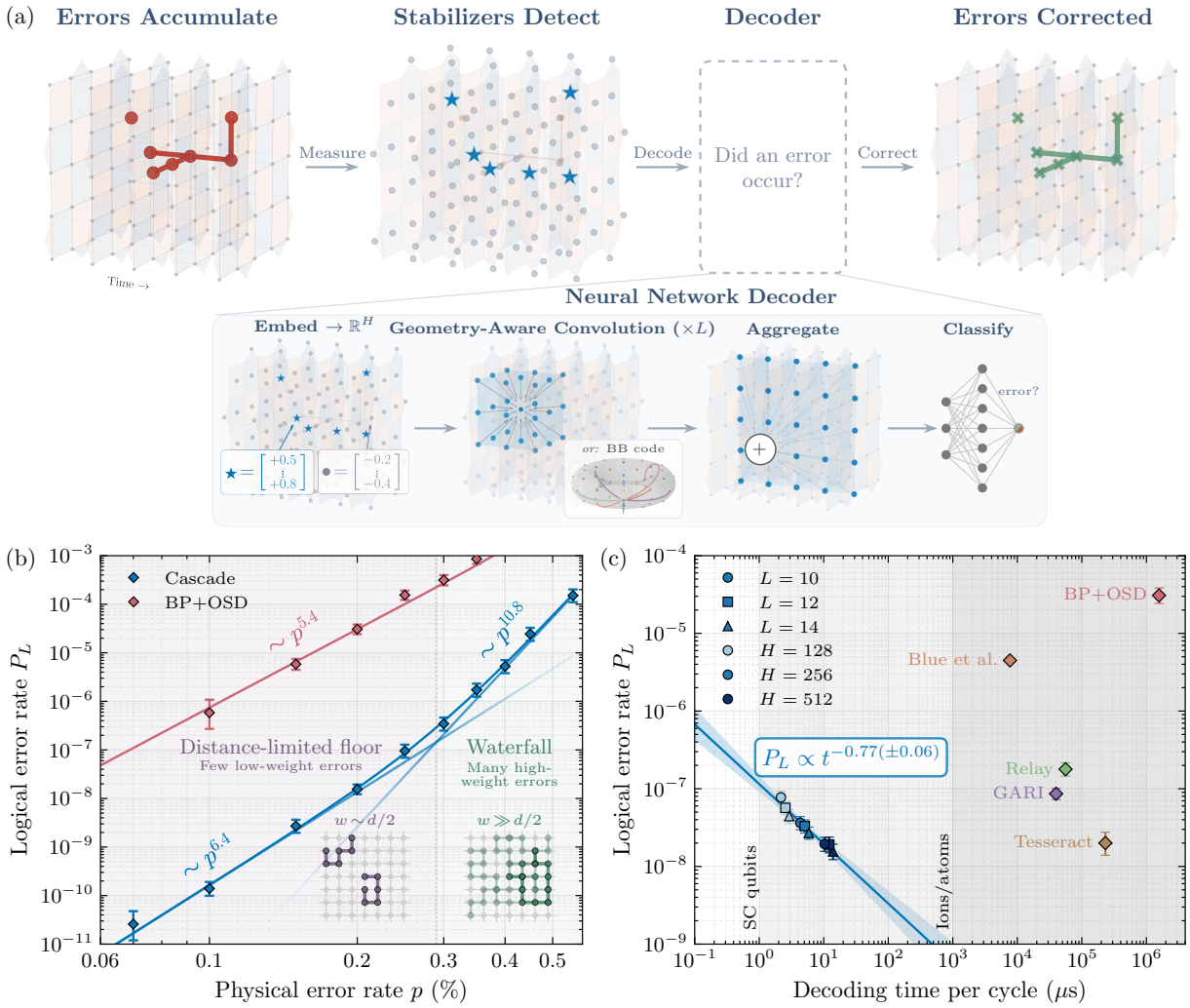

Figure 1. **Cascade: structure-aware neural decoding.** (a) *Pipeline.* Binary detection events from repeated stabilizer measurements are embedded per-site into an H-dimensional representation, processed by L convolutional layers whose weights are indexed by the code's translation group (3D lattice for surface codes; torus $\mathbb{Z}_\ell \times \mathbb{Z}_m$ for bivariate bicycle codes), scattered to the data-qubit sites of each logical operator's support, and mapped to per-observable probabilities by a small prediction head. The top row shows the quantum-error-correction context: physical errors corrupt data qubits; stabilizer measurements extract a spacetime syndrome; the decoder predicts whether a logical error occurred. (b) *Error suppression on the* $[\![144, 12, 12]\!]$ *Gross code* ($R = d$ rounds, circuit-level depolarizing noise). The logical error rate per cycle exhibits two regimes: a steep waterfall ($\sim p^{10.8}$) dominated at moderate p by numerous high-weight failure modes, and a distance-limited floor ($\sim p^{6.4}$) emerging at very low noise. BP+OSD (orange, $\sim p^{5.4}$) misses the waterfall entirely. (c) *Accuracy–latency tradeoff at* $p = 0.2\%$. Cascade (GPU inference on NVIDIA H200) spans a range of amortized latencies while achieving lower logical error rates than prior decoders (single-threaded CPU). Diamond markers show reported single-shot latencies of BP+OSD (Roffe et al., 2020), Relay (Müller et al., 2025), Tesseract (Beni et al., 2025), and the transformer decoder of (Blue et al., 2025). Error bars: 95% credible intervals.

on periodic lattices), Cascade's convolutional operations are equivariant to the code's translation group, realizing direction-dependent learned message passing (Gilmer et al., 2017; Veličković & Blundell, 2021) in place of BP's fixed update rules, in the spirit of group-equivariant and geometric deep learning (Cohen & Welling, 2016; Bronstein et al., 2021). Three empirical findings follow. First, a capacity-dependent transition: error suppression improves monotonically with hidden width H and saturates near $H \approx 64$, above which the decoder matches or exceeds the best hand-designed baselines. Second, robustness to distribution shift: models trained at a single high noise level remain accurate and well-calibrated (Guo et al., 2017) across seven orders of magnitude in logical error rate. Third, a physics finding enabled by the first two: in this waterfall regime (Richardson & Urbanke, 2001; 2009), P_L drops far more steeply with p than the distance-limited scaling $P_L \sim p^{\lfloor (d+1)/2 \rfloor}$ used in current qubit-count estimates (Gidney & Ekerå, 2021; Litinski, 2019; Beverland et al., 2022). For the $[\![144, 12, 12]\!]$ Gross code, the two regimes are cleanly separated: a waterfall term $P_L \sim p^{11}$ and a distance-limited floor $P_L \sim p^{6.4}$ (Figure 1(b)), implying substantially lower space-time overhead for fault-tolerant quantum computation than standard resource estimates predict.

2. Problem setup: decoding as structured prediction

A stabilizer code (Gottesman, 1997), denoted $[\![n, k, d]\!]$, encodes k logical qubits into $n \gg k$ physical qubits with minimum distance d; an error on physical qubits corrupts the logical state only if the error pattern is *topologically nontrivial* with respect to the code (i.e., cannot be undone by any sequence of local corrections). Information about errors is extracted non-destructively by measuring m commuting Pauli operators (the *stabilizers*) every cycle. Differences between consecutive measurement rounds give *detection events*, and over R rounds form a binary tensor $s \in \{0,1\}^{R \times m}$, the *syndrome*. The decoding task is to predict, from s, a binary vector $\ell \in \{0,1\}^k$ indicating which of the k logical observables has flipped: structured binary classification on a spatiotemporal graph whose connectivity is determined by the code's parity-check structure. The code's *distance* d is the minimum weight of any error that produces a trivial syndrome but a nontrivial logical flip; below a noise-model-dependent threshold p_{th}, the logical error rate can in principle be suppressed exponentially with d.

Two code families structure the experiments below. Surface codes (Fowler et al., 2012) arrange stabilizers on a 2D grid, so the spacetime syndrome lives on a 3D lattice; BB codes (Bravyi et al., 2024) arrange them on a torus $\mathbb{Z}_\ell \times \mathbb{Z}_m$, giving a syndrome on (torus $\times$ time). We evaluate under *circuit-level* depolarizing noise: each gate, idle, and measurement operation in the stabilizer-extraction circuit is independently depolarized at rate p, so errors propagate through the circuit before becoming visible in the syndrome. This is the standard hardware-realistic benchmark and is considerably more challenging than the simpler *data-level* model (depolarizing errors applied only to data qubits between rounds), which we use in the capacity-scaling study of Section 4.2 for training efficiency. We report logical error rate per logical qubit per cycle P_L. We compare against five decoders: belief propagation (BP) (Poulin & Chung, 2008); BP with ordered-statistics post-processing (BP+OSD) (Roffe et al., 2020); minimum-weight perfect matching (MWPM) (Higgott, 2022), applicable only to surface codes; Relay (Müller et al., 2025), which augments BP with learned corrections; and Tesseract (Beni et al., 2025), a near-optimal combinatorial search whose $\sim$1 s-per-shot cost precludes real-time use but serves as an accuracy benchmark.

3. Structure-aware neural decoding

Design principles. Cascade is built around three inductive biases that match the structure of stabilizer codes on regular lattices. *Locality*: errors produce spatially localized syndrome patterns that can be progressively resolved by layers of local processing. *Translation equivariance*: every stabilizer has the same local connectivity up to a lattice translation, so the same decoding rules should apply at every site. *Anisotropy*: information arriving from different directions carries distinct meaning; a surface code stabilizer's horizontal and diagonal neighbors are of different type (X versus Z), and the temporal direction encodes measurement errors rather than data-qubit errors. Belief propagation satisfies only locality: its fixed update rules treat the graph symmetrically, which is exactly what causes its convergence failures on quantum codes (Poulin & Chung, 2008; Raveendran & Vasić, 2021).

Group-equivariant convolution. Let G denote the code's translation group: $G = \mathbb{Z}^3$ for the spacetime lattice of a surface code, or $G = \mathbb{Z}_\ell \times \mathbb{Z}_m \times \mathbb{Z}$ for a BB code on the torus $\mathbb{Z}_\ell \times \mathbb{Z}_m$. Stabilizer codes in general have multiple site types (e.g., X-checks and Z-checks, and for BB codes two further data-qubit types), so G acts on each site-type orbit separately. Cascade's convolutional weights $W_\delta^{(\ell)}$ are indexed by the relative offset $\delta \in G$ between source and target sites (with a separate set of weights per source $\to$ target site-type pair, suppressed below for clarity), giving the update rule

$$h_v^{(\ell+1)} = \sum_{u \in \mathcal{N}(v)} W_{\delta(u,v)}^{(\ell)} \, h_u^{(\ell)}, \tag{1}$$

Algorithm 1 Cascade forward pass

Input: syndrome $s \in \{0,1\}^{R\times m}$; group G; kernel size K; depth L; width H; bottleneck factor b=4
$z \leftarrow \mathrm{Embed}(s) \in \mathbb{R}^{R\times m\times H}$
for $\ell = 1, \ldots, L$ **do**
  // pre-activation bottleneck residual block
  $u \leftarrow \mathrm{PW}^{(\ell)}_{H\to H/b}(\mathrm{SiLU}(\mathrm{BN}(z)))$
  $u \leftarrow \mathrm{GroupConv}^{(\ell)}_{G,K}(\mathrm{SiLU}(\mathrm{BN}(u)))$ {G-equivariant spatial}
  $u \leftarrow \mathrm{PW}^{(\ell)}_{H/b\to H}(\mathrm{SiLU}(\mathrm{BN}(u)))$
  $z \leftarrow z + u$
end for
$\tilde{z} \leftarrow \mathrm{ScatterConv}(z;$ data-qubit map$)$
for $o = 1, \ldots, k$ **do**
  $\hat{p}_o \leftarrow \sigma\big(\mathrm{MLP}_{2H}\big(\mathrm{Pool}\big(\tilde{z}[\mathrm{supp}(\bar{o})]\big)\big)\big)$
end for
Output: per-observable probabilities $\{\hat{p}_o\}_{o=1}^{k}$

where $\mathcal{N}(v)$ includes v itself. Because W depends only on the relative offset (and type pair), not on the absolute position, layers are G-equivariant on each orbit in the sense of (Cohen & Welling, 2016; Bronstein et al., 2021): translating the syndrome by $g \in G$ translates every internal representation by the same g. In contrast to typical applications of geometric deep learning, the equivariance group G here is not a modeling hyperparameter; it is fully determined by the code's physical structure. For surface codes this reduces to a standard 3D convolution with a small number of type channels; for BB codes it generalizes to indices in the torus group, giving a direction- and type-dependent kernel that respects the code's non-trivial lattice symmetries. With L such layers each of hidden width H, the network realizes learned, anisotropic message passing (Gilmer et al., 2017; Veličković & Blundell, 2021) whose receptive field grows with depth; once $L \sim d$, it integrates information across the full code distance, resolving local errors first and global topological ambiguities last.

Forward pass. Algorithm 1 summarizes the pipeline. After L equivariant convolutions on syndrome sites, representations are scattered to data qubits through a fixed, code-specified map, pooled over the support of each logical operator, and mapped to per-observable logit by a small MLP. The decoder outputs a probability $\hat{p}_o \in [0,1]$ for each of the k logical observables; training minimizes binary cross-entropy between $\hat{p}_o$ and the true flip ℓ_o, end-to-end, at a single high physical error rate p_{train}. No auxiliary losses, curriculum, or labeled data at target evaluation noise levels are required.

Scope. The equivariance framework applies to any stabilizer code whose checks share a common local structure up to a lattice symmetry, including color codes (Bombin & Martin-Delgado, 2006), toric codes, hyperbolic surface codes (Breuckmann & Eberhardt, 2021), and other quantum LDPC codes on periodic lattices or Cayley graphs. For codes with less regularity, stabilizers can be grouped into equivalence classes that share weights, reducing to a standard graph neural network (Gilmer et al., 2017) in the fully irregular limit. Our largest model (used for BB code experiments below, $L = 14$, $H = 512$) has on the order of 10^8 parameters, modest by modern ML standards and, as the capacity-scaling experiment below shows, already above the threshold at which near-optimal decoding emerges. Full hyperparameters, training schedule, and parameter counts across all code sizes are listed in Section B.

Relation to prior equivariant decoders. The idea of building a code's symmetry into a neural decoder was introduced by the END decoder (Egorov et al., 2023), a translation-equivariant CNN that predicts the logical class of the toric code under code-capacity (data-level) noise. Cascade shares this inductive bias but extends it along the two axes that govern practical decoding: it operates on the full circuit-level depolarizing syndrome, where errors propagate through the extraction circuit and the symmetry group acquires a temporal factor ($G = \mathbb{Z}_\ell \times \mathbb{Z}_m \times \mathbb{Z}$), and it applies the same equivariant template to high-rate quantum LDPC codes such as the bivariate bicycle codes rather than the toric code alone. These extensions are what expose the waterfall regime and reach $P_L \sim 10^{-10}$ on the Gross code, beyond the reach of code-capacity, single-code equivariant decoders.

Relation to AlphaQubit. The closest prior neural decoder is AlphaQubit (Bausch et al., 2024; Senior et al., 2025), a transformer-based architecture that achieves near-optimal accuracy on surface and color codes up to distances 23 and 27, ingests analog readout signals, handles leakage, reaches logical error rates below 10^{-10} per cycle, and decodes experimental data from Google's quantum processors. AlphaQubit and Cascade are independent, complementary efforts.[1] Architecturally, AlphaQubit's global self-attention imposes no geometric assumptions and applies to any code topology, while Cascade trades that generality for an $O(n)$ convolutional structure tailored to codes with translation symmetry, an inductive bias that lets it extend naturally to bivariate bicycle codes and other high-rate quantum LDPC

[1] A head-to-head numerical comparison is further limited by a scope difference in evaluation: AlphaQubit targets Google's superconducting-hardware noise model (analog I/Q readout, leakage, device-specific circuit-level noise), whereas Cascade is evaluated under standard circuit-level depolarizing noise, the common benchmark in the QEC decoder literature. Extending Cascade to hardware-realistic noise with analog soft inputs, leakage handling, and real-hardware validation is an important direction for neural decoders of this class.

codes with analogous lattice symmetry. Empirically, both decoders, trained at a single elevated physical error rate, extrapolate reliably to logical error rates many orders of magnitude lower. That this appears across very different architectures (transformer vs. convolution) and different code families (surface/color vs. BB) suggests it is a robust feature of sufficiently expressive neural decoders, and argues against a common concern that learned decoders might harbor error floors invisible at training noise levels.

4. Results

4.1. Error suppression on BB and surface codes

Gross code. On the $[\![144, 12, 12]\!]$ Gross code under circuit-level depolarizing noise, our largest model (L=14, H=512) achieves logical error rates that are $\sim$4000$\times$ below BP+OSD (Roffe et al., 2020) and $\sim$17$\times$ below Relay (Müller et al., 2025) at $p = 0.1\%$, with accuracy comparable to the combinatorial Tesseract decoder (Beni et al., 2025).[2] Below threshold, the logical error rate drops far more steeply than the distance bound $P_L \sim p^{\lfloor (d+1)/2 \rfloor}$ would predict (Figure 1(b)): we find a clean two-regime fit with a steep "waterfall" term $\sim p^{11}$ dominating at moderate p, and a distance-limited floor $\sim p^{6.4}$ emerging only at very low noise. A more powerful decoder corrects a larger fraction of low-weight errors, exposing this steep regime; BP+OSD misses it entirely ($\sim p^{5.4}$). A combinatorial account of minimal failure-mode counts $N(w)$, which underlies the two-regime structure, is given in Section A. No error floor is observed down to $P_L \approx 2 \times 10^{-11}$.

BB codes. Figure 2(a–c) extends this analysis to three BB codes of increasing distance ($d = 6, 12, 18$). Cascade consistently outperforms the reference decoders across all three sizes; on the largest code ($[\![288, 12, 18]\!]$), it reaches $P_L \sim 10^{-10}$ per logical qubit per cycle at $p = 0.2\%$, into the regime required for large-scale algorithms. AlphaQubit (Senior et al., 2025) is the closest neural baseline but was evaluated on surface codes under a different noise model, so a direct head-to-head comparison on BB codes is not possible; the transformer decoder of (Blue et al., 2025), which shares our noise model on BB codes, is included in Figs. 1(c) and 2(d) for reference.

Surface codes. On surface codes at $p = 0.2\%$, fitting $P_L \propto \Lambda^{-\lfloor (d+1)/2 \rfloor}$ over $d = 7$ to 19 (Figure 3), Cascade yields $\Lambda \approx 8.4$, compared to 5.0 for uncorrelated MWPM, 7.8 for correlated MWPM, and 9.1 for Tesseract. The resource-estimation projections of (Gidney & Ekerå, 2021; Litinski, 2019; Beverland et al., 2022) assume $\Lambda \approx 10$ at $p = 0.1\%$ (calibrated to MWPM-class decoders); a decoder that exceeds this value at the same physical error rate reduces the code distance required to reach a target logical error rate. Concretely, Cascade reaches $P_L \sim 10^{-9}$ at distance d=15, compared to d=19 for MWPM (a $\sim$40% reduction in physical qubit count), and this advantage grows as targets tighten toward the 10^{-10}–10^{-12} regime of algorithmic fault tolerance.

Latency. These accuracy gains are only relevant if the decoder is fast enough for real-time use. On a single NVIDIA H200 GPU, our models run at $\sim$40 μs per cycle single-shot; batched inference reaches amortized latencies up to two orders of magnitude lower, for 3,000–100,000$\times$ higher throughput than existing decoders running single-threaded on CPU.[3] Which figure of merit matters depends on the setting: single-shot latency constrains mid-circuit measurements that gate subsequent operations, while amortized latency is what matters for memory experiments or deep Clifford circuits where rounds can be decoded in parallel (Terhal, 2015; Skoric et al., 2023). These latencies are well within the $\sim$1 ms budgets of trapped-ion and neutral-atom platforms (Bluvstein et al., 2024), though above the $\sim$1 μs required for superconducting qubits; roofline estimates suggest FLOP-efficient variants on dedicated hardware could approach that tighter budget (Section C).

4.2. Capacity scaling

How much of a code's error-correcting capability a decoder realizes depends strongly on its capacity. We sweep hidden width H for surface-code models trained at a single noise level (Figure 4) and measure the resulting error-suppression exponent m in $P_L \propto p^m$. Small models ($H \lesssim 64$) suppress errors worse than uncorrelated MWPM; as H grows, m rises continuously and saturates at $m \approx 8$ for $H \gtrsim 64$, near the optimal distance-bounded value. This capacity-dependent transition gives a first-principles explanation for why existing hand-designed decoders miss the waterfall regime: they simply lack the representational capacity to recognize the complex, higher-weight error patterns that dominate at moderate p. The transition is sharper under circuit-level noise, where optimal performance includes the waterfall regime itself (Section A).

4.3. Calibration under distribution shift

Models trained at a single high noise level p_{train} remain accurate *and* well-calibrated across physical error rates spanning seven orders of magnitude in logical error rate (Figure 5(c)). Neural networks routinely lose calibration under even mild distribution shift (Guo et al., 2017), yet here

[2] BP+OSD (Roffe et al., 2020), Relay (Müller et al., 2025), and Tesseract (Beni et al., 2025) are each run with their authors' recommended default settings.

[3] Reported amortized latencies are total wall-clock time divided by (batch size $\times$ syndrome rounds).

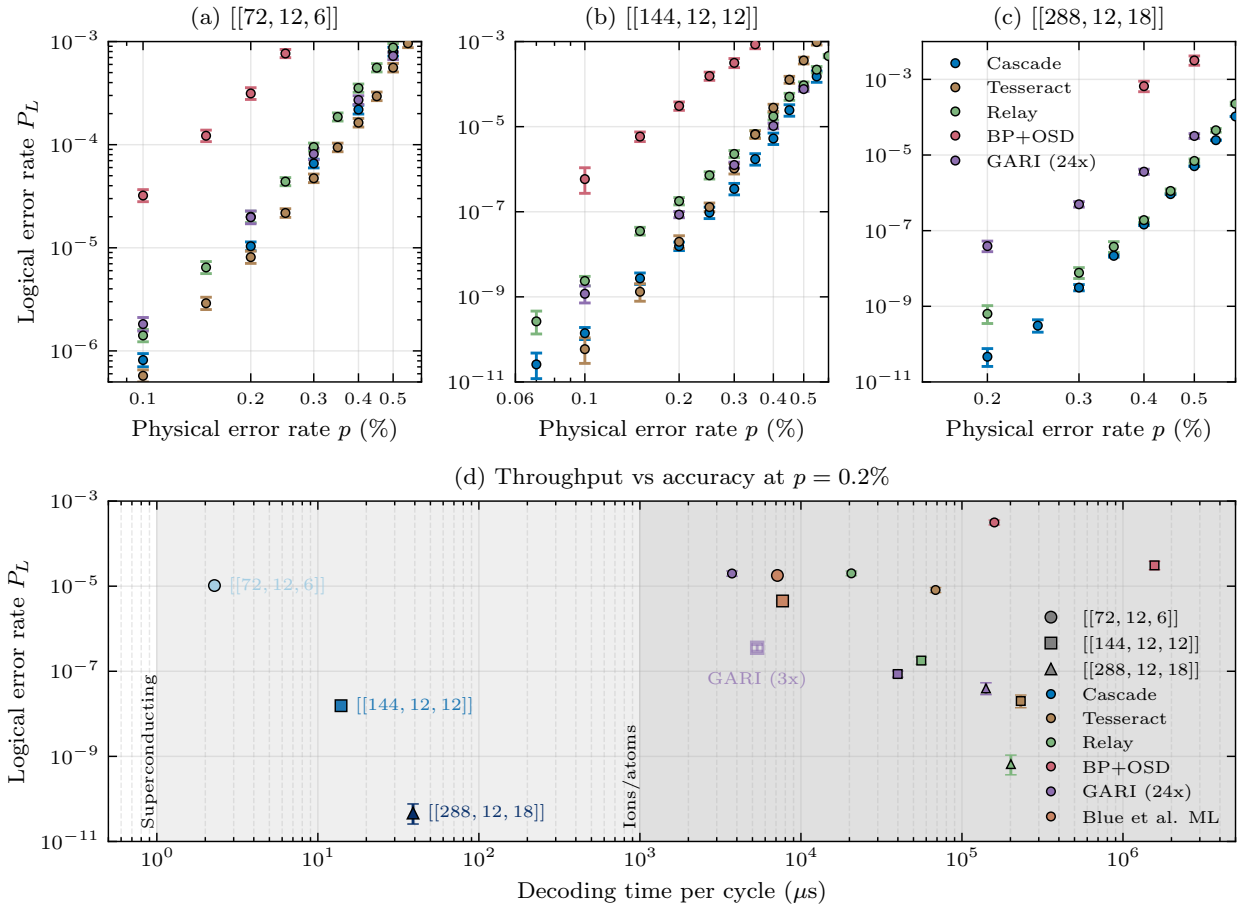

Figure 2. **Distance scaling of BB code decoders under circuit-level depolarizing noise.** (a–c) Logical error rate per logical qubit per round ($R = d$ rounds) versus physical error rate for the $[\![72, 12, 6]\!]$, $[\![144, 12, 12]\!]$, and $[\![288, 12, 18]\!]$ bivariate bicycle codes, respectively. Cascade achieves lower logical error rates than BP+OSD, Relay, and (where evaluable) Tesseract across all code sizes. (d) Accuracy vs. latency at $p = 0.2\%$ for all three BB codes (timing reported for BB codes is amortized latency). We include the published results of a different ML decoder (Blue et al., 2025) for the $[\![72, 12, 6]\!]$ and $[\![144, 12, 12]\!]$ codes. For the $[\![288, 12, 18]\!]$ code, we are unable to evaluate Tesseract due to its computational cost; BP+OSD is similarly intractable at lower noise levels.

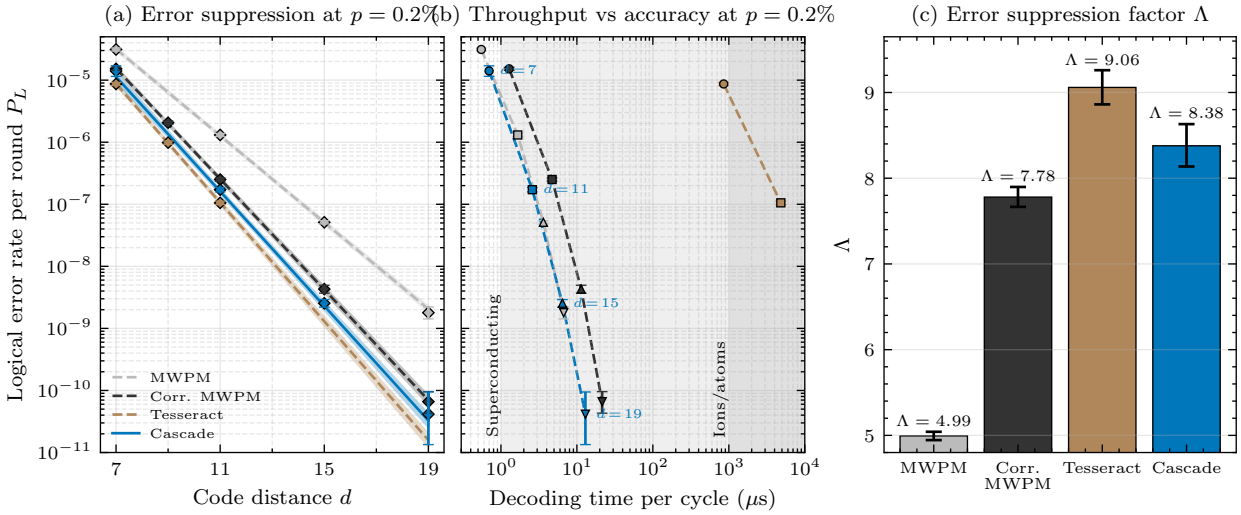

Figure 3. **Distance scaling of surface code decoders at $p = 0.2\%$ under circuit-level depolarizing noise.** (a) Logical error rate per round versus code distance d for MWPM, correlated MWPM, Tesseract, and Cascade. All decoders exhibit exponential error suppression ($P_L \propto \Lambda^{-\lfloor (d+1)/2 \rfloor}$), with Cascade and Tesseract achieving the steepest slopes. (b) Accuracy–latency trade-off across code distances (GPU inference on NVIDIA H200). Amortized latency (dashed lines, batched inference) is significantly lower than the single-shot latency (solid lines, unbatched inference). (c) Error suppression factor Λ extracted from exponential fits to panel (a).

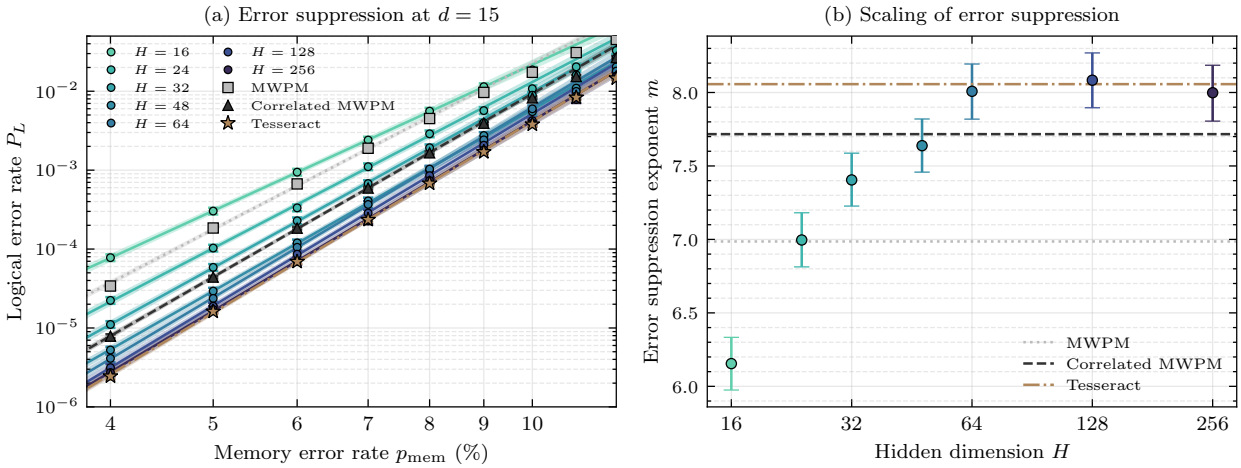

Figure 4. **Error suppression improves with model capacity.** (a) Logical error rate versus memory error rate p_{mem} at distance $d = 15$ for surface code models with varying hidden dimension H (data-level depolarizing noise, used here for training efficiency). Larger models achieve lower error rates across the full range of noise levels. (b) Error suppression exponent m for memory errors (from fitting $P_L \propto p^m$) versus hidden dimension. Small models ($H \lesssim 64$) exhibit poor scaling exponents well below reference decoders (MWPM, correlated matching), while large models ($H \geq 64$) approach optimal performance.

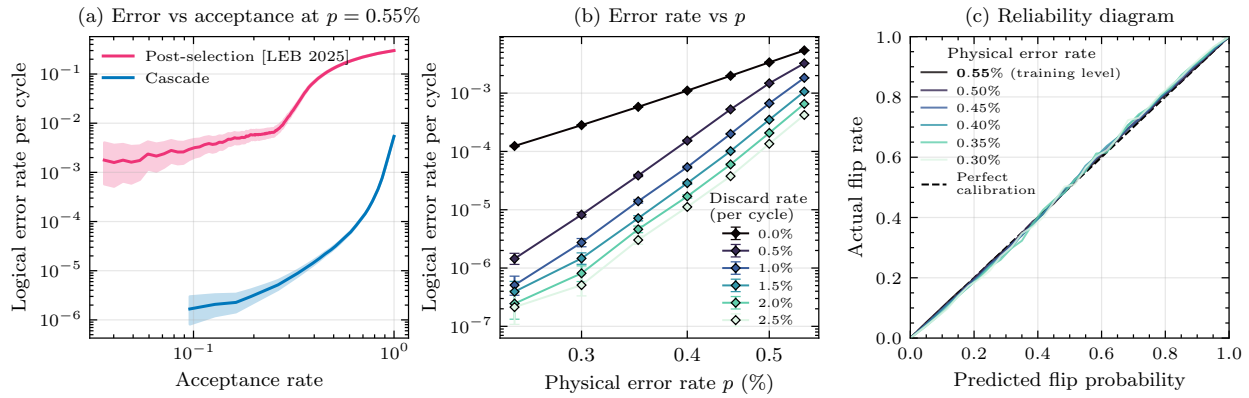

Figure 5. **Confidence-aware decoding with the ML decoder on the $[\![72, 12, 6]\!]$ bivariate bicycle code.** (a) Logical error rate per cycle as a function of acceptance rate at $p = 0.55\%$. Our decoder (blue) achieves much steeper error suppression (as a function of acceptance rate) compared to cluster-based post-selection methods (Lee et al., 2025) (pink). (b) Logical error rate versus physical error rate for different discard rates. (c) Reliability diagram showing calibration across physical error rates (color scale); alignment with the diagonal indicates well-calibrated predictions.

the predicted logical-flip probabilities track the empirical rates across the entire evaluation sweep, without any temperature scaling or calibration data at evaluation noise levels. Calibration enables post-selection: by discarding low-confidence predictions we trade acceptance rate for accuracy (Figure 5(a)). On the $[\![72, 12, 6]\!]$ code at $p = 0.55\%$, Cascade reaches $P_L \sim 2 \times 10^{-3}$ with $\sim$95% acceptance, compared to $\sim$5% for cluster-based post-selection (Lee et al., 2025). Many fault-tolerant subroutines (e.g., magic-state and entanglement distillation (Campbell et al., 2017; Smith et al., 2024; Zhou et al., 2025; Menon et al., 2025)) proceed by attempting an operation, discarding low-confidence outcomes, and retrying; acceptance rate therefore sets the effective runtime, so the $\sim$20$\times$ higher acceptance directly reduces time overhead. Even a 0.5% per-cycle discard rate yields up to two orders-of-magnitude reduction in P_L (Figure 5(b)).

5. Discussion and outlook

These results have several implications for fault-tolerant quantum computation. The waterfall has remained largely undercharacterized for quantum codes under realistic circuit-level noise, precisely because existing decoders lacked the accuracy to expose it.

First, code design and resource estimation should move beyond distance as the sole figure of merit. The sparsity of low-weight uncorrectable errors, the weight distribution of logical operators, and decoder accuracy all shape effective error suppression and can differ dramatically between codes with identical $[\![n, k, d]\!]$ parameters. Code-specific models that capture these structural properties will yield more favorable resource estimates than the standard distance-based formula.

Second, the decoder should be treated as an integral component of fault-tolerant architecture co-design rather than an independent subsystem. The capacity scaling results (Figure 4) show a clear threshold: below it, decoders cannot represent the higher-weight error patterns that drive the waterfall; above it, near-optimal performance emerges. Decoder power therefore directly determines how much of a code's error-correcting capability can be realized in practice, and existing decoders miss the waterfall because they sit below this threshold.

Third, our decoder's geometric inductive bias (locality, translation equivariance, and anisotropy) generalizes across codes: the same architecture achieves near-optimal accuracy on surface codes and bivariate bicycle codes, code families that have historically required entirely different decoding strategies (Higgott & Gidney, 2025; Roffe et al., 2020; Maurer et al., 2025; Fowler, 2013). Since these biases apply to any code with a common local check structure, we expect Cascade to extend to other high-rate qLDPC families (Xu et al., 2024; Kasai, 2026; Panteleev & Kalachev, 2021), where the waterfall should be even more pronounced at larger code sizes. The architecture's local, feed-forward, deterministic-latency structure is also well-suited to FPGA/ASIC deployment (Section C).

Finally, trapped-ion (Ballance et al., 2016), superconducting (Li et al., 2024), and neutral-atom (Evered et al., 2025) hardware has now reached entangling error rates near $\sim$0.1%, the regime where the waterfall delivers logical error rates sufficient for practical algorithms.

References

AMD. Versal architecture and product data sheet: Overview (DS950). https://docs.amd.com/v/u/en-US/

ds950-versal-overview, 2024.

Ballance, C. J., Harty, T. P., Linke, N. M., Sepiol, M. A., and Lucas, D. M. High-fidelity quantum logic gates using trapped-ion hyperfine qubits. *Physical Review Letters*, 117(6):060504, aug 2016. ISSN 1079-7114. doi: 10.1103/physrevlett.117.060504. URL http://dx.doi.org/10.1103/PhysRevLett.117.060504.

Bausch, J., Senior, A. W., Heras, F. J. H., Edlich, T., Davies, A., Newman, M., Jones, C., Satzinger, K., Niu, M. Y., Blackwell, S., Holland, G., Kafri, D., Atalaya, J., Gidney, C., Hassabis, D., Boixo, S., Neven, H., and Kohli, P. Learning high-accuracy error decoding for quantum processors. *Nature*, 635(8040):834–840, 11 2024. ISSN 0028-0836. doi: 10.1038/s41586-024-08148-8.

Beni, L. A., Higgott, O., and Shutty, N. Tesseract: A search-based decoder for quantum error correction. 3 2025. URL http://arxiv.org/abs/2503.10988.

Beverland, M. E., Murali, P., Troyer, M., Svore, K. M., Hoefler, T., Kliuchnikov, V., Low, G. H., Soeken, M., Sundaram, A., and Vaschillo, A. Assessing requirements to scale to practical quantum advantage. 11 2022. URL http://arxiv.org/abs/2211.07629.

Blue, J., Avlani, H., He, Z., Ziyin, L., and Chuang, I. L. Machine learning decoding of circuit-level noise for bivariate bicycle codes. 4 2025. URL http://arxiv.org/abs/2504.13043.

Bluvstein, D., Evered, S. J., Geim, A. A., Li, S. H., Zhou, H., Manovitz, T., Ebadi, S., Cain, M., Kalinowski, M., Hangleiter, D., Ataides, J. P. B., Maskara, N., Cong, I., Gao, X., Rodriguez, P. S., Karolyshyn, T., Semeghini, G., Gullans, M. J., Greiner, M., Vuletić, V., and Lukin, M. D. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2 2024. ISSN 0028-0836. doi: 10.1038/s41586-023-06927-3.

Bombin, H. and Martin-Delgado, M. A. Topological quantum distillation. *Physical Review Letters*, 97(18):180501, 2006. doi: 10.1103/PhysRevLett.97.180501.

Bravyi, S., Cross, A. W., Gambetta, J. M., Maslov, D., Rall, P., and Yoder, T. J. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, 3 2024. ISSN 0028-0836. doi: 10.1038/s41586-024-07107-7.

Breuckmann, N. P. and Eberhardt, J. N. Quantum low-density parity-check codes. *PRX Quantum*, 2(4), 10 2021. ISSN 2691-3399. doi: 10.1103/prxquantum.2.040101.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Campbell, E. T., Terhal, B. M., and Vuillot, C. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 9 2017. ISSN 0028-0836. doi: 10.1038/nature23460.

Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., Lu, Y., and Le, Q. V. Symbolic discovery of optimization algorithms. 2 2023. URL http://arxiv.org/abs/2302.06675.

Cohen, T. S. and Welling, M. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48, pp. 2990–2999, 2016.

Egorov, E., Bondesan, R., and Welling, M. The end: An equivariant neural decoder for quantum error correction, 2023. URL https://arxiv.org/abs/2304.07362.

Evered, S. J., Geim, A. A., Bluvstein, D., Kalinowski, M., Manovitz, T., Zhou, H., Li, S. H., Maskara, N., and Lukin, M. D. High-fidelity entangling gates and nonlocal circuits with neutral atoms. *In preparation*, 2025.

Fowler, A. G. Optimal complexity correction of correlated errors in the surface code. 10 2013. URL http://arxiv.org/abs/1310.0863.

Fowler, A. G., Mariantoni, M., Martinis, J. M., and Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 9 2012. ISSN 1050-2947. doi: 10.1103/PhysRevA.86.032324.

Gidney, C. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, 7 2021. ISSN 2521-327X. doi: 10.22331/q-2021-07-06-497.

Gidney, C. and Ekerå, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 4 2021. ISSN 2521-327X. doi: 10.22331/q-2021-04-15-433.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pp. 1263–1272, 2017.

Gottesman, D. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pp. 1321–1330, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Higgott, O. PyMatching: A Python Package for Decoding Quantum Codes with Minimum-Weight Perfect Matching. *ACM Transactions on Quantum Computing*, 3(3):1–16, 9 2022. ISSN 2643-6809. doi: 10.1145/3505637.

Higgott, O. and Gidney, C. Sparse Blossom: correcting a million errors per core second with minimum-weight matching. *Quantum*, 9:1600, 1 2025. ISSN 2521-327X. doi: 10.22331/q-2025-01-20-1600.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. 3 2018. URL http://arxiv.org/abs/1803.05407.

Jordan, K., Jin, Y., Boza, V., Jiacheng, Y., Cesista, F., Newhouse, L., and Bernstein, J. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., luc Cantin, P., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., and Yoon, D. H. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Computer Architecture News*, 45(2):1–12, 6 2017. ISSN 0163-5964. doi: 10.1145/3140659.3080246.

Kasai, K. Breaking the orthogonality barrier in quantum ldpc codes. 1 2026. URL http://arxiv.org/abs/2601.08824.

Kasai, K., Hagiwara, M., Imai, H., and Sakaniwa, K. Quantum error correction beyond the bounded distance decoding limit. *IEEE Transactions on Information Theory*, 58(2):1223–1230, feb 2012. ISSN 1557-9654. doi: 10.1109/tit.2011.2167593. URL http://dx.doi.org/10.1109/TIT.2011.2167593.

Komoto, D. and Kasai, K. Quantum error correction near the coding theoretical bound. 12 2025. URL http://arxiv.org/abs/2412.21171.

Lee, S.-H., English, L., and Bartlett, S. D. Efficient post-selection for general quantum ldpc codes. 10 2025. URL http://arxiv.org/abs/2510.05795.

Leverrier, A. and Zemor, G. Quantum Tanner codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 872–883. IEEE, 10 2022. doi: 10.1109/FOCS54457.2022.00117.

Li, R., Kubo, K., Ho, Y., Yan, Z., Nakamura, Y., and Goto, H. Realization of high-fidelity cz gate based on a double-transmon coupler. *Physical Review X*, 14(4):041050, nov 2024. ISSN 2160-3308. doi: 10.1103/physrevx.14.041050. URL http://dx.doi.org/10.1103/PhysRevX.14.041050.

Litinski, D. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, 3 2019. ISSN 2521-327X. doi: 10.22331/q-2019-03-05-128.

Maurer, T., Bühler, M., Kröner, M., Haverkamp, F., Müller, T., Vandeth, D., and Johnson, B. R. Real-time decoding of the gross code memory with FPGAs. 10 2025. URL http://arxiv.org/abs/2510.21600.

Menon, V., Bonilla-Ataides, J. P., Mehta, R., Gu, A., Tan, D. B., and Lukin, M. D. Magic tricycles: Efficient magic state generation with finite block-length quantum ldpc codes. 8 2025. URL http://arxiv.org/abs/2508.10714.

Müller, T., Alexander, T., Beverland, M. E., Bühler, M., Johnson, B. R., Maurer, T., and Vandeth, D. Improved belief propagation is sufficient for real-time decoding of quantum memory. 6 2025. URL http://arxiv.org/abs/2506.01779.

Panteleev, P. and Kalachev, G. Degenerate Quantum LDPC Codes With Good Finite Length Performance. *Quantum*, 5:585, 11 2021. ISSN 2521-327X. doi: 10.22331/q-2021-11-22-585.

Panteleev, P. and Kalachev, G. Quantum ldpc codes with almost linear minimum distance. *IEEE Transactions on Information Theory*, 68(1):213–229, 2022. doi: 10.1109/TIT.2021.3119384.

Poulin, D. and Chung, Y. On the iterative decoding of sparse quantum codes. *Quantum Information and Computation*, 8(10):986–1000, 11 2008. ISSN 1533-7146. doi: 10.26421/QIC8.10-8.

Raveendran, N. and Vasić, B. Trapping sets of quantum ldpc codes. *Quantum*, 5:562, 10 2021. ISSN 2521-327X. doi: 10.22331/q-2021-10-14-562.

Richardson, T. J. and Urbanke, R. L. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2): 599–618, 2 2001. ISSN 0018-9448. doi: 10.1109/18.910577.

Richardson, T. J. and Urbanke, R. L. *Modern coding theory*. Cambridge University Press, Cambridge, 2009.

Roffe, J., White, D. R., Burton, S., and Campbell, E. Decoding across the quantum low-density parity-check code landscape. *Physical Review Research*, 2(4):043423, 12 2020. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.2.043423.

Senior, A. W., Edlich, T., Heras, F. J. H., Zhang, L. M., Higgott, O., Spencer, J. S., Applebaum, T., Blackwell, S., Ledford, J., Žemgulytė, A., Žídek, A., Shutty, N., Cowie, A., Li, Y., Holland, G., Brooks, P., Beattie, C., Newman, M., Davies, A., Jones, C., Boixo, S., Neven, H., Kohli, P., and Bausch, J. A scalable and real-time neural decoder for topological quantum codes. 12 2025. URL http://arxiv.org/abs/2512.07737.

Skoric, L., Browne, D. E., Barnes, K. M., Gillespie, N. I., and Campbell, E. T. Parallel window decoding enables scalable fault tolerant quantum computation. *Nature Communications*, 14(1):7040, 11 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-42482-1.

Smith, S. C., Brown, B. J., and Bartlett, S. D. Mitigating errors in logical qubits. *Communications Physics*, 7(1), 11 2024. ISSN 2399-3650. doi: 10.1038/s42005-024-01883-4.

Terhal, B. M. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307–346, 4 2015. ISSN 0034-6861. doi: 10.1103/RevModPhys.87.307.

torchao. Torchao: Pytorch-native training-to-serving model optimization. https://github.com/pytorch/ao, 2024. BSD-3-Clause license.

Veličković, P. and Blundell, C. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021. doi: 10.1016/j.patter.2021.100273.

Watson, F. H. E. and Barrett, S. D. Logical error rate scaling of the toric code. *New Journal of Physics*, 16(9):093045, sep 2014. ISSN 1367-2630. doi: 10.1088/1367-2630/16/9/093045. URL http://dx.doi.org/10.1088/1367-2630/16/9/093045.

Xu, Q., Bonilla Ataides, J. P., Pattison, C. A., Raveendran, N., Bluvstein, D., Wurtz, J., Vasić, B., Lukin, M. D., Jiang, L., and Zhou, H. Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *Nature Physics*, 20(7):1084–1090, jul 2024. doi: 10.1038/s41567-024-02479-z.

Yang, G., Hu, E., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097. Curran Associates, Inc., 2021.

Zhou, Z., Pexton, S., Kubica, A., and Ding, Y. Error mitigation of fault-tolerant quantum circuits with soft information. 12 2025. URL http://arxiv.org/abs/2512.09863.

A. Failure-mode combinatorics and the waterfall

Here we justify the two-regime fit used in Section 4.1. For any stabilizer code and decoder, a *minimal failure mode* of weight w is a set of w independent fault locations whose simultaneous activation causes a logical failure, with no failing proper subset. Under circuit-level depolarizing noise at physical error rate p, the probability of activating exactly a given weight-w subset is $p^w(1-p)^{n-w}$ to leading order, and the logical error rate decomposes as

$$P_L \approx \sum_{w \geq w_{\min}} N(w)\, p^w, \tag{2}$$

where $N(w)$ counts the minimal failure modes of weight w and $w_{\min} = \lfloor (d+1)/2 \rfloor$ is the nominal distance bound. The standard resource-estimation scaling $P_L \sim p^{w_{\min}}$ assumes that the $w = w_{\min}$ term dominates at the physical error rates of interest. For the codes studied here, $N(w)$ is instead heavily concentrated at $w \gg w_{\min}$: the code admits relatively few minimal failure modes at the distance bound but many more at higher weights, so the product $N(w)\, p^w$ is dominated by higher-weight terms across a wide range of p below threshold, giving a suppression exponent much larger than $w_{\min}$. Only at very low p does the steeply-decaying $p^{w_{\min}}$ eventually outweigh the multiplicity and set a distance-limited floor.

The set of minimal failure modes depends on the decoder: a more accurate decoder suppresses more of the low-weight modes, exposing the regime in which the higher-weight multiplicity dominates. Decoders like BP+OSD fail on too many low-weight errors, so the steep regime is never exposed; on the Gross code, BP+OSD exhibits only $\sim p^{5.4}$ scaling rather than the $\sim p^{11}$ waterfall accessible to Cascade. The two-regime structure is familiar in classical LDPC coding (Richardson & Urbanke, 2001; 2009); indications of similar behavior have appeared in quantum codes (Watson & Barrett, 2014; Kasai et al., 2012; Komoto & Kasai, 2025), but existing decoders lack the accuracy to resolve it in the practically relevant regime below threshold.

On surface codes, the same mechanism manifests as a decoder-dependent error-suppression factor Λ in $P_L \propto \Lambda^{-\lfloor (d+1)/2 \rfloor}$. At $p = 0.2\%$, across distances $d = 7$ to 19, we measure $\Lambda \approx 5.0$ for uncorrelated MWPM, $\Lambda \approx 7.8$ for correlated MWPM, $\Lambda \approx 8.4$ for Cascade, and $\Lambda \approx 9.1$ for Tesseract. The nearly two-fold spread exceeds what threshold differences alone can account for and reflects each decoder's differing ability to correctly handle the higher-weight error patterns that drive the waterfall.

B. Architecture and training details

Backbone. Cascade's backbone is a stack of L bottleneck residual blocks (He et al., 2016) operating at hidden dimension H. Each block projects $H \to H/4$, applies the code-specific convolution, projects back $H/4 \to H$, and adds a residual skip. Each projection and convolution is preceded by batch normalization and a SiLU activation. Depth scales with distance as $L \sim d$ so that the receptive field covers the full code; the largest configurations used are $L = 14$, $H = 512$ for the $[\![288, 12, 18]\!]$ BB code and comparable sizes for large-d surface codes (the latter converge in ∼200 H200 GPU-hours; the former in <100). To train stably across widths we use Maximal Update Parameterization (MuP) (Yang et al., 2021).

Code-specific convolution. For surface codes, the convolution is a standard 3D convolution with kernel $3 \times 3 \times 3$ (two spatial, one temporal). For BB codes, the check-to-check connectivity is dictated by the Tanner graph: a full check-to-check step has 22 spatial neighbors across 3 temporal offsets (66 learned relations per layer); we factor it as check→data→check, reducing to 6 spatial neighbors and 2 temporal offsets (12 relations, a $> 5\times$ reduction). We implement these factorized convolutions using custom Triton kernels that exploit the regular torus structure for efficient memory access. The update at position v is

$$h_v^{(\ell+1)} = \sum_{u \in \mathcal{N}(v)} W^{(\ell)}_{\delta(u,v)}\, h_u^{(\ell)} + W^{(\ell)}_{\text{self}}\, h_v^{(\ell)}, \tag{3}$$

where $\delta(u, v)$ is the relative offset in the code's translation group, and $W^{(\ell)}_{\delta}$ depends only on δ, not on the absolute position: the hallmark of a group-equivariant convolution.

Readout. After the last block, a final convolution scatters syndrome-site representations to data qubits; the data-qubit representations in each logical operator's support are average-pooled and passed through a 2-layer MLP with hidden dimension $2H$ to produce a per-observable logit.

Training. Syndrome/label pairs are generated on-the-fly with Stim (Gidney, 2021) under circuit-level depolarizing noise, $R = d$ rounds per sample, at a single training noise level ($p_{\text{train}} = 0.7\%$ for surface codes; $p_{\text{train}} = 0.55\%$ for BB codes). The loss is binary cross-entropy averaged across logical observables. We optimize matrix-valued parameters with Muon (Jordan et al., 2024) (peak lr 3×10^{-3}) and scalars with Lion (Chen et al., 2023) (peak lr 2×10^{-4}), following a cosine schedule over 50,000 steps (1000-step warmup, decaying to $1/10$ of peak) and continuing training for a total of 80,000 steps, weight decay 3×10^{-3}, batch size 3328, bfloat16 mixed precision with gradient-norm clipping. All reported results use an exponential moving average of weights ($\beta = 0.9998$) (Izmailov et al., 2018). Models typically converge within 3×10^8 training examples. A brief three-stage noise-level curriculum ($p_1 \to p_2$ annealing for $\leq 2\%$ of total steps) prevents the prolonged random-output plateau observed when training directly at p_{train}.

Capacity scaling experiment. For Figure 4 we train surface-code decoders of fixed depth $L = 8$ and varying width H at $d = 15$ under data-level depolarizing noise at $p = 13\%$, using data-level noise for training efficiency across the many configurations.

C. Roofline estimates and computational cost

A deployed decoder running inside a quantum computer's control stack must meet a hardware-dependent latency budget per syndrome round: $\sim$1 ms for trapped-ion and neutral-atom platforms, $\sim$1 μs for superconducting qubits. Our GPU measurements ($\sim$40 μs/cycle single-shot on an H200) meet the former comfortably and miss the latter by roughly an order of magnitude. Here we estimate what FLOP-efficient variants on dedicated hardware would achieve.

MAC counts. Each bottleneck block with hidden dimension H, bottleneck factor b, kernel size K, and n syndrome positions costs

$$\begin{aligned} \text{Convolution:} \quad & \sim 2nH^2/b + nK(H/b)^2, \\ \text{Depthwise conv.:} \quad & \sim 2nH^2/b + nK(H/b), \\ \text{Local attention:} \quad & \sim 2nH^2/b + nK(H/b) + 3n(H/b)^2, \\ \text{Full attention:} \quad & \sim 2nH^2/b + n^2(H/b) + 3n(H/b)^2. \end{aligned} \tag{4}$$

MACs are the atomic hardware primitive on ASICs/FPGAs: one MAC corresponds to one multiplier circuit, so MAC counts directly proxy chip area, power, and latency.

Post-training quantization. Cascade tolerates post-training quantization from FP32 to FP8 with *no* measurable accuracy loss on any code or noise level, using `torchao` (torchao, 2024) after folding batch-normalization parameters into the preceding convolution weights. Since MAC area scales roughly quadratically with bit width, this yields a $\sim$16$\times$ reduction in multiplier area and power. Combined with the $\sim$4$\times$ reduction from the depthwise-convolution variant, we estimate a compound $\sim$60$\times$ reduction in multiplier area over an FP32 non-depthwise baseline. Further reduction to INT4 via quantization-aware training is a natural next step.

Per-round latency. Figure 6 shows roofline latency per syndrome round as a function of hidden dimension H for both code families, assuming the peak throughput of an AMD Versal AI Core FPGA (133 TOPS INT8 (AMD, 2024)); FP8 and INT8 throughput are comparable on modern tensor engines. A TPU v1 (92 TOPS) (Jouppi et al., 2017) or Edge TPU (4 TOPS) would shift all curves upward by $1.4\times$ or $33\times$. Even the largest models sit well within the $\sim$1 ms budget of trapped-ion and neutral-atom platforms. For superconducting qubits ($\sim$1 μs), standard convolutions at $H = 256$ require $\sim$10 μs per round, roughly an order of magnitude too slow; depthwise convolutions at moderate widths ($H \leq 128$) approach the target, and further reduction via INT4 or smaller models could close the remaining gap. BB codes achieve lower latency than surface codes at equal H because both n and L are smaller ($n = 144$ versus $d^2 = 361$ at $d = 19$).

These are roofline figures assuming 100% hardware utilization. On GPUs, single-sample (unbatched) inference is dominated by kernel launch overhead and memory transfers rather than compute, making GPU latency a poor proxy for dedicated hardware performance. On FPGAs and ASICs, where the computation graph can be spatially mapped and weights stored on-chip, utilization approaching the roofline is achievable for regular, feed-forward architectures like ours.

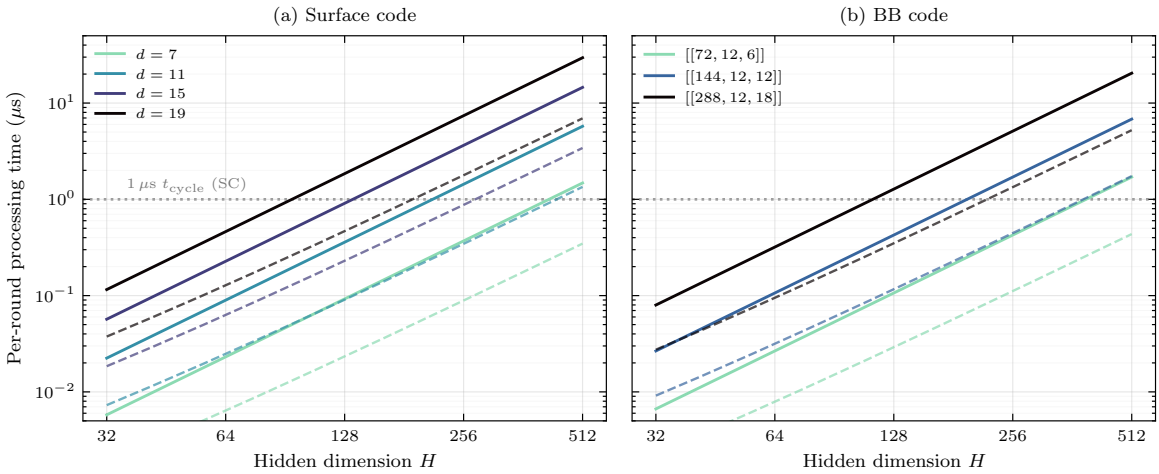

Figure 6. **Roofline latency per syndrome round on an AMD Versal AI Core FPGA (133 TOPS INT8).** Solid lines: standard convolution; dashed lines: depthwise convolution. Both panels assume network depth $L = d$ and bottleneck factor $b = 4$. (a) Surface codes with $K = 27$ ($3 \times 3 \times 3$ kernel) and $n = d^2$ detectors per round. (b) Bivariate bicycle codes with $K = 24$ (two bipartite steps of 12 neighbors each) and n equal to the number of physical qubits. The horizontal dotted line marks the 1 μs-per-round budget for superconducting qubits.